Electronic Version
Stylesheet Version v1.1.1

Description

Reducing Number of Computations in a Neural Network Modeling Several Data Sets

Background of Invention

[0001] A typical neural network contains 'neurons', which are connected by 'weights'. Weights generally serve to communicate the results/conclusions reached by one neuron to other neurons. The weights are adjusted while examining a data set until the neural network models the system consistent with the data set (within an acceptable error level). In the share price example above, the weights of the neural network and the operation of neurons are adjusted until the share price of each day (in a data set) is predicted based on the price of the prior days (in the data set).

[0002] A prior neural network may be designed to start with random values for weights, and adjust the values iteratively while examining each data set. Various approaches may

be used for such adjustment and example approaches are described in a document entitled, "An Introduction to Neural Networks", available at the URL: http://www.cs.stir.ac.uk/~lss/NNIntro/InvSlides.html.

[0003] One problem with starting with random weights for each system is that it may require a large number of iterations to determine weights, which would model a system at a desired level of accuracy. The resulting required large number of computations and large amount of time may be unacceptable in several environments. Accordingly, there is a general need to reduce the number of computations while modeling systems using neural networks.

BRIEF DESCRIPTION OF DRAWINGS

[0004] The present invention will be described with reference to the accompanying drawings briefly described below.

[0005] Figure (Fig.) 1 is a diagram depicting the various elements of an example neural network Figure 2 is a block diagram illustrating the details of an approach to implementing a neural network according to various aspects of the present invention.

[0006] Figure 3 is a flow chart illustrating the manner in which the initial weights of a neural network may be determined according to an aspect of the present invention.

[0007] Figure 4 is a flow chart illustrating the manner in which a determination can be made as to whether two data sets follow similar pattern, in one embodiment.

[0008] Figure 5 is a block diagram illustrating the details of an example embodiment implementing various aspects of the present invention in the form of software instructions.

[0009] In the drawings, like reference numbers generally indicate identical, functionally similar, and/or structurally similar elements. The drawing in which an element first appears is indicated by the leftmost digit(s) in the corresponding reference number.

DETAILED DESCRIPTION

[0010] *1. Overview*

[0011] An aspect of the present invention reduces the number of computations required to model a system characterized by a data set. To achieve such a reduction, a system checks whether there exists another system, which has been already modeled as a neural network and data set characterizing the another system follows a similar pattern as the data set characterizing the system sought to be modeled. If such an another system exists, the weights computed for the another system are used as the initial

weights for the system sought to be modeled.

[0012] Due to such initial weights, the final weight values for the system sought to be modeled may be computed in a fewer number of iterations (compared to a prior approach in which the weights are initialized to random values). Accordingly, the computational requirements may be reduced at least in some cases.

[0013] In an embodiment, similarity of data sets is checked by using a "curve fitting" technique to fit each data set into a respective polynomial, and using least square method to determine the distance between the two data sets. If the distance is less than a threshold (which can be either pre-computed or determined dynamically), data sets may be deemed to be of similar pattern.

[0014] According to another aspect of the present invention, weights corresponding to pre-modeled data sets are stored in a data storage (such as database). The weights may thus be retrieved and used as initial weights for systems sought to be characterized later. The data sets and/or computed polynomials (in general, data characterizing the prior behavior of the system) may also be stored, which enables checking of similarity of patterns of data sets.

[0015] Several aspects of the invention are described below with reference to examples for illustration. It should be understood that numerous specific details, relationships, and methods are set forth to provide a full understanding of the invention. One skilled in the relevant art, however, will readily recognize that the invention can be practiced without one or more of the specific details, or with other methods, etc. In other instances, well-known structures or operations are not shown in detail to avoid obscuring the invention.

[0016] *2. Sample Neural Network*

[0017] Figure 1 is a diagram illustrating the details of an example neural network in which various aspects of the present invention can be implemented. The neural network is shown containing four layers namely input layer 101, hidden layers 102-1 and 102-2, and output layer 103. Each layer and the corresponding components are described below in detail.

[0018] Input layer 101 is shown containing neurons 104-1 through 104-n. Each neuron 104-1 through 104-n may receive one or more elements of a data set, and process the corresponding elements to generate one or more weights. The weights are shown communicated to the

neurons in the next layer (102-1) as represented by the lines below reference numeral 108.

[0019] Hidden layer 102-1 is shown containing neurons 105-1 to 105-p, which are shown receiving weights from input layer 101, and generating weights for the subsequent layer 102-2. Similarly, hidden layer 102-2 is shown containing neurons 106-1 through 106-Q, which receive weights from prior layers and generate corresponding weights.

[0020] Output layer 103 may also contain one or more neurons, which operate according to the weights received from the prior layer to generate the desired outputs.

[0021] From the above, it may be appreciated that each neuron operates based on the weights received from various other neurons. In general, the weights are sought to be adjusted using various approaches until the neural network appropriately models the prior data points. The models thus generated can be used, for example, to predict future results.

[0022] As noted above, a prior approach may start with random values for weights while modeling each system, which may lead to a large number of iterations. The description is continued with an illustration of an example implemen-

tation of a neural network according to various aspects of the present invention, which reduces such large number of iterations at least in some cases.

[0023] *3. Example implementation*

[0024] Figure 2 is a block diagram representing a logical view in which a neural network may be implemented according to various aspects of the present invention. Neural network 200 of Figure 2 is shown containing input blocks 210-1 to 210-n, initial weights determination block 250, weights computation block 260, storage block 270 and output block 280. Each block is described below in further detail.

[0025] Each of input blocks 210-1 to 210-n may receive one or more data elements (e.g., the daily closing stock share prices when the stock price is sought to be modeled on a daily basis) of a data set sought to be modeled. Each data element can be represented in various forms, e.g., vectors, polar coordinates, as appropriate for the specific system. The received data elements may be forwarded to initial weights determination block 250.

[0026] Output block 280 receives the weights for a present iteration and the data elements based on which the output values are to be generated, computes the corresponding outputs, and provides the computed outputs to weights

computation block 260. The weights and data elements may be received from weights computation block 260.

[0027] Weights computation block 260 determines the weights of the neural network for each successive iteration based on the output values received from output block 280. The weights to be used in a first iteration may be received from initial weights determination block 250. Weights computation block 260 may provide the determined weights and the specific data elements to be used in computing the output values for a present iteration to output block 280.

[0028] The weights may be adjusted using various techniques well known in the relevant arts by comparing the received output values with the expected output values. The weights are adjusted until the data set is modeled within a desired degree of accuracy.

[0029] The final weights thus computed to model the received data set may be stored in storage block 270. Storage block 270 may contain non-volatile storage (data storage) for storing the final computed weights as well as the data sets and/or computed polynomials (in general, data characterizing the behavior of the system).

[0030] Initial weights determination block 250 provides initials

weights to be used in a first iteration. The initial weights thus provided may lead to reduced number of iterations or computations in weights computation block 260. The manner in which the initial weights may be determined according to an aspect of the present invention is described below with reference to Figure 3.

[0031] *4. Method*

[0032] Figure 3 is a flow chart illustrating the manner in which the initial weights of a neural network may be determined according to various aspects of the present invention. The flow chart is described with reference to Figure 2 merely for illustration. The flow chart begins in step 301 and control immediately passes to step 310.

[0033] In step 310, input blocks 210-1 through 210-n receive a first data set characterizing the behavior of a first system to be modeled. The data set may contain one or more data elements, as noted above.

[0034] In step 330, neural network 200 (or weight computation block 250 and output block 280 together) models the first system based on the data elements contained in the first data set. Such modeling may be performed using any of prior approaches as well as various aspects of the present invention. A first set of weights may be determined due to

such modeling.

[0035] In step 340, weights computation block 260 stores the first set of weights in storage block 270. In addition, the first data set or any other data (e.g., the coefficients of polynomials, described below with reference to Figure 4) characterizing the behavior of the first system may also be stored.

[0036] In step 350, input blocks 210-1 through 210-n receive a second data set characterizing a second system sought to be modeled. The second data set may also contain multiple data elements.

[0037] In step 370, initial weight determination block 250 may determine if the data elements in the second data set follow a similar pattern as the data elements in the first data set. An example approach to determine such similarity is described below with reference to Figure 4. However, other approaches can be used to determine similarity. Control passes to step 380 if a similarity is detected, and to step 390 otherwise.

[0038] In step 380, neural network 200 models the second data set using the first set of weights as initial weights. Due to such use of first set of weights, the number of computations required for step 380 may be reduced (in compari-

son with a prior approach when only random weights are used as initial weights). Control then passes to step 395.

[0039] In step 390, neural network 200 models the second data set starting with random initial weights (or other values determined using other approaches). Control then passes to step 395.

[0040] In step 395, weights determined corresponding to the second data set may also be stored in storage block 270. Data characterizing the behavior of the second system may also be stored in storage block 270. The stored information may be used again in steps 370 and 380, as appropriate, while modeling additional systems to reduce the computational requirements. The method ends at step 399.

[0041] While the above approach(es) are described with reference to only two data sets merely for illustration, it should be understood that the approaches can be easily extended to use many data sets/systems over time. The description is continued with respect to the manner in which a determination can be made as to whether two data sets follow a similar pattern.

[0042] *5. Determining Whether Data Sets Follow a Similar Pattern*

[0043] Figure 4 is a flow chart illustrating the manner in which a

determination can be made as to whether two data sets follow similar pattern, in one embodiment. The flow chart of figure 4 begins at step 401 and control immediately passes to step 420.

[0044] In step 420, initial weight determination block 250 fits the first data set into a first curve having a first set of coefficients. For ease of comparison (in step 470), the data elements may be normalized to a pre-specifed range (e.g., 0 to 1 on a linear scale), and the normalized data elements may be fit into a curve. Such curve fitting may be performed in a known way.

[0045] In one embodiment, curve fitting is performed using 'Least Square Method' technique, which helps in ascertaining the line or curve of "best" fit for a set of data points. According to this technique, data points are plotted on a graph and a smooth line is drawn through the midst of them and a distance of each point from a corresponding point on the line/curve may be computed as difference between the observed and predicted results.

[0046] The line/curve of best fit is that for which the average of squares of these distances is least (or within an acceptable error limit). For illustration, it is assumed that the first data set is fit into a cubical function of ($a1\ x^3 + b1\ x^2 +$

c1 x + d1), wherein a1, b1, c1 and d1 represent the coefficients.

[0047] In step 430, initial weight determination block 250 fits the second data set (or normalized data element values) also into a second curve. For illustration, it is assumed that the second data set is fit into a cubical function of (a2 x^3 + b2 x^2 + c2 x + d2), wherein a2, b2, c2 and d2 represent the coefficients for the second cubical function.

[0048] In step 450, the distance between the first set of coefficients and the second set of coefficients is computed. In the illustrative example, the distance may be computed using a formula (sqrt ($(a2-a1)^2+(b2-b1)^2+(c2-cb1)^2+(d2-d1)^2$)). Instead of normalizing the data elements of the data sets, in an alternative embodiment, the coefficients may be normalized prior to computation of the distance.

[0049] In step 470, initial weight determination block 250 determines if the distance is less than a threshold value. If the distance is less than the threshold value, control transfers to step 480 else to step 490.

[0050] In step 480, initial weight determination block 250 determines that the two sets of data points follow similar pattern, and control passes to step 499. In step 490, initial

weight determination block 250 determines that the two sets of data points do not follow similar pattern, and control passes to step 499. The method ends at step 499.

[0051] The features described above may be attained by appropriate design/implementation of software instructions executing on a digital processing system, as described below in further detail.

[0052] *6. Software-driven Implementation*

[0053] Figure 5 is a block diagram illustrating the details of how various aspects of the invention may be implemented substantially in the form of software in an embodiment of the present invention. System 500 may contain one or more processors such as central processing unit (CPU) 510, random access memory (RAM) 520, secondary memory 530, graphics controller 560, display unit 570, network interface 580, and input interface 590. All the components except display unit 570 may communicate with each other over communication path 550, which may contain several buses as is well known in the relevant arts. The components of Figure 5 are described below in further detail.

[0054] CPU 510 may execute instructions stored in RAM 520 to provide several features of the present invention. For ex-

ample, determination of initial weight and modeling of systems using neural network, may be performed due to such execution. CPU 510 may contain multiple processing units, with each processing unit potentially being designed for a specific task. Alternatively, CPU 510 may contain only a single general purpose-processing unit. RAM 520 may receive instructions from secondary memory 530 using communication path 550.

[0055] Graphics controller 560 generates display signals (e.g., in RGB format) to display unit 570 based on data/instructions received from CPU 510. Display unit 570 contains a display screen to display the images defined by the display signals. Input interface 590 may correspond to a keyboard and/or mouse. Graphics controller 560 and input interface 590 may enable a user to interact directly with system 500.

[0056] Secondary memory 530 may contain hard drive 535, flash memory 536 and removable storage drive 537. Secondary memory 530 may store the data and software instructions, which enable system 500 to provide several features in accordance with the present invention. Some or all of the data and instructions may be provided on removable storage unit 540, and the data and instructions

may be read and provided by removable storage drive 537 to CPU 510. Floppy drive, magnetic tape drive, CD-ROM drive, DVD Drive, Flash memory, removable memory chip (PCMCIA Card, EPROM) are examples of such removable storage drive 537.

[0057] Removable storage unit 540 may be implemented using medium and storage format compatible with removable storage drive 537 such that removable storage drive 537 can read the data and instructions. Thus, removable storage unit 540 includes a computer readable storage medium having stored therein computer software and/or data.

[0058] In this document, the term "computer program product" is used to generally refer to removable storage unit 540 or hard disk installed in hard drive 535. These computer program products are means for providing software to system 500. CPU 510 may retrieve the software instructions, and execute the instructions to provide various features of the present invention as described above.

[0059] *7. Conclusion*

[0060] While various embodiments of the present invention have been described above, it should be understood that they have been presented by way of example only, and not

limitation. Thus, the breadth and scope of the present invention should not be limited by any of the above-described example embodiments, but should be defined only in accordance with the following claims and their equivalents.